IFW AF

PTO/SB/17 (12/04)
Approved for use through 09/30/2005. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2005

Patent fees are subject to annual revision.
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12.
See 37 C.F.R. §§ 1.27 AND 1.28

TOTAL AMOUNT OF PAYMENT	($) 0.00

Complete if Known	
Application Number	09/483,723
Filing Date	January 14, 2000
First Named Inventor	Sharon S. Liu
Examiner Name	James A. Reagan
Group/Art Unit	3621
Attorney Docket No.	15437-0109

METHOD OF PAYMENT (check one)

1. ☒ Throughout the pendency of this application, please charge any additional fees, including any required extension of time fees, and credit all overpayments to deposit account 50-1302. A duplicate of this sheet is enclosed.

Deposit Account Number: 50-1302

Deposit Account Name: Hickman Palermo Truong & Becker, LLP

2. □ Payment Enclosed:
   □ Check □ Money Order □ Other

3. □ Applicant(s) is entitled to small entity status. See 37 CFR 1.27.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1011	300	2011	150	Utility filing fee	
1111	500	2111	250	Utility Search fee	
1311	200	2311	100	Utility Examination fee	
1081	250	2081	125	Utility Application Size Fee	
1005	200	2005	100	Provisional Application Fee	
1085	250	20835	125	Provisional Application Size Fee	
				SUBTOTAL (1)	($) 0.00

2. EXTRA CLAIM FEES

	Highest Paid Claims		Extra Claims		Fee from Below		Fee Paid
Total Claims	75	-78	0	X	50.00	=	0.00
Independent Claims	3	- 3=	0	X	200.00	=	0.00
Multiple Dependent						=	

***or number previously paid, if greater; For Reissues, see below*

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	**Reissue independent claims over original patent
1205	50	2205	25	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge – late provisional filing fee or cover sheet.	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1504	300	2504	300	Publication Fee	
1462	400	1462	400	Petitions Director not specifically provided for Group I	
1463	200	1463	200	Petitions Director not specifically provided for Group II	
1464	130	1464	130	Petitions Director not specifically provided for Group III	
1806	180	1806	180	Submission of information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	

Other fee (specify) ____________________

Other fee (specify) ____________________

*Reduced by Basic Filing Fee Paid SUBTOTAL (3) ($) 0.00

SUBMITTED BY

Name (Print/Type)	Christian A. Nicholes	*Registration No. (Attorney/Agent)*		*Telephone*	(408) 414-1080
Signature				*Date*	June 7, 2005

WARNING: Information on this form may become public. Credit card information should not be Included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Amend, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Confirmation No.: 8755

Sharon S. Liu, et al. Group Art Unit No.: 3621

Serial No.: 09/483,723 Examiner: James A. Reagan

Filed: January 14, 2000

For: OBJECT ORIENTED MECHANISM FOR DYNAMICALLY CONSTRUCTING CUSTOMIZED IMPLEMENTATIONS TO ENFORCE RESTRICTIONS

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on January 28, 2005, and corrects and replaces the allegedly defective Appeal Brief that was filed on March 10, 2005.

I. REAL PARTY IN INTEREST

Sun Microsystems, Inc. is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

U.S. Patent Application Serial No. 09/483,724 is related to the present application and is currently under appeal before the Board of Patent Appeals and Interferences.

III. STATUS OF CLAIMS

Claims 11-23, 34-46, and 57-105 are pending in this application, have been finally rejected, and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendments were filed after the Office Action mailed on October 28, 2004.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claims 11, 34, and 57. These independent claims recite similar limitations, except in the context of a method, a framework, and a computer-readable medium, respectively. Claims 11, 34, and 57 are directed generally to an object-oriented approach for dynamically constructing customized implementations to enforce restrictions on services. For purposes of the present invention, a service is defined broadly to encompass any functionality requested by and provided to an application, including but not limited to encryption/decryption functionality.

According to the approach recited in Claims 11, 34, and 57, a framework receives, from an application, a request for an implementation of a particular service, such as an implementation of a particular encryption algorithm. In response, the framework determines what restrictions, if any, need to be imposed on the requested implementation. Once the restrictions are determined, the framework dynamically constructs the requested implementation. The requested implementation is constructed such that it incorporates the general (i.e. unrestricted) implementation of the service, the restrictions, and enforcement logic for enforcing the restrictions on the general implementation. Since the requested implementation is constructed specifically for the application, it is customized for the application. Thus, the implementation is referred to as the customized implementation.

The framework instantiates a wrapper instance as a part of dynamically constructing the customized implementation. The wrapper instance comprises enforcement logic for enforcing the restrictions. The framework then encapsulates both the restrictions and an instance of the general implementation within the wrapper instance. The wrapper instance ensures that the restrictions are enforced.

Once the encapsulation process is complete, the framework provides the newly constructed wrapper instance to the application that requested the customized implementation. The wrapper instance is provided as the customized implementation requested by the application. Due to the mapping of the methods of the wrapper instance to the proper methods of the general implementation instance, the application may invoke the methods of the wrapper instance in the same manner as the application would have invoked the methods of the general implementation instance. Consequently, the application does not need to be modified to handle the wrapper instance differently than the general implementation instance.

Since the customized implementation incorporates the restrictions and enforcement logic for enforcing the restrictions, it is not necessary for the application to further interact with the framework. The customized implementation itself will provide the services, and will guarantee that the restrictions are enforced. By dynamically constructing customized implementations in this manner, the framework ensures that the necessary restrictions are enforced on the services provided to the application

(Specification, at page 2, lines 23-25, page 3, lines 1-25, page 14, lines 6-25, page 15, lines 1-4, FIG. 2, and FIGS. 4A-B).

More specifically, Claim 11 recites (with reference annotations in parenthesis):

In a system (FIG. 1, 100) comprising an application (FIG. 1, 104), a framework (FIG. 1, 102) and an implementation class (FIG. 1, 106) which provides an implementation for a particular service, a method performed by the framework, comprising:

receiving (FIG. 4A, 404; page 10, line 24 to page 11, line 6) a request from an application for a customized implementation of a particular service;

instantiating (FIG. 4A, 428; page 12, lines 4-7) an implementation class (page 10, lines 2-6) which provides an implementation for the particular service (page 6, lines 16-18) to give rise to an implementation instance;

determining (FIG. 4A, 436; page 12, lines 14-24) a set of zero or more restrictions to be imposed on said customized implementation;

instantiating (FIG. 4B, 448; page 14, lines 12-13) a wrapper class (FIG. 3A, 306) to give rise to a wrapper instance, said wrapper instance comprising enforcement logic (page 15, line 21 to page 16, line 4) for enforcing said restrictions;

encapsulating (FIG. 4B, 452; page 14, lines 13-16) said implementation instance and said restrictions within said wrapper instance; and

providing (FIG. 4B, 456; page 15, lines 1-4) said wrapper instance to the application as said customized implementation;

wherein said wrapper instance comprises one or more invocable methods (FIG. 3A, 306; page 14, lines 16-25), wherein said implementation instance comprises one or more invocable methods (FIG. 3B, 312; page 14, lines 16-25), and wherein encapsulating comprises:

mapping (page 14, lines 16-25) the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance.

Independent Claims 34 and 57 recite corresponding features described at the same locations in the application. The dependent claims are not argued separately from the independent claims upon which they depend.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 11, 12, 15, 18, 19, 34, 35, 38, 41, 42, 57, 58, 61, 64, 65, 70-80, 83-89, 92-98, and 101-105 stand rejected under 35 U.S.C. §103(a) as being allegedly unpatentable over Taylor, "Object-Oriented Information Systems" ("Taylor") in view of Pressman, "Software Engineering: A Practitioner's Approach." ("Pressman").

2. Claims 13, 14, 16, 17, 20-23, 36, 37, 39, 40, 43-46, 59, 60, 62, 63, 66-69, 81, 82, 90, 91, 99, and 100 stand rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Taylor in view of Pressman and Shanton, U.S. Patent No. 5,369,702 ("Shanton").

VIII. ARGUMENTS

A. Introduction

It is well founded that to establish a *prima facie* case of obviousness under 35 U.S.C. § 103(a), the references cited and relied upon must teach or suggest all the claim limitations.

With respect to the present application, it is respectfully submitted that Taylor and Pressman, considered alone or in combination, do not teach or suggest all the limitations of Claims 11, 12, 15, 18, 19, 34, 35, 38, 41, 42, 57, 58, 61, 64, 65, 70-80, 83-89, 92-98, and 101-105.

It is also respectfully submitted that Taylor, Pressman, and Shanton, considered alone or in combination, do not teach or suggest all the limitations of Claims 13, 14, 16, 17, 20-23, 36, 37, 39, 40, 43-46, 59, 60, 62, 63, 66-69, 81, 82, 90, 91, 99, and 100.

B. The Limitations of Claims 11, 12, 15, 18, 19, 34, 35, 38, 41, 42, 57, 58, 61, 64, 65, 70-80, 83-89, 92-98, and 101-105 Are Not in Any Way Taught or Suggested by Taylor and Pressman

Claim 11 addresses the problem of how to ensure that restrictions are enforced on an implementation of a service that is plugged in to a framework. According to the approach recited in Claim 11, the framework receives, from an application, a request for a customized implementation of a particular service. The framework instantiates an implementation class. The implementation class provides an implementation for the particular service. The instantiation of the implementation class gives rise to an implementation instance, which comprises one or more invocable methods. The framework determines a set of restrictions to be imposed on the customized implementation.

The framework instantiates a wrapper class. The wrapper class comprises enforcement logic for enforcing the restrictions. The instantiation of the wrapper class gives rise to a wrapper instance, which also comprises one or more invocable methods. The framework encapsulates the implementation instance and the restrictions within the wrapper instance. In doing so, the framework maps the invocable methods of the wrapper instance to the invocable methods of the implementation instance. The framework provides the wrapper instance to the application as the customized implementation. Because the wrapper instance comprises logic for enforcing restrictions, and because the restrictions are encapsulated within the wrapper instance along with the implementation instance, the customized implementation will provide the particular service, and restrictions on that particular service will be enforced.

Summary of Examiner's Contentions

In rejecting Claim 11, the Examiner contends that Taylor and Pressman disclose the above features. The Examiner alleges that Taylor discloses a wrapper instance on page 296. The Examiner asserts that the rest of the features of Claim 11 are merely object-oriented programming and design principles, and are common knowledge within the art. The Examiner asserts that at least some of these principles are disclosed in Taylor, Pressman, or both. However, with the exception of the "wrapper instance," the Examiner does not specifically indicate the locations at which each feature of Claim 11 is allegedly disclosed, or even which reference allegedly discloses that feature.

Appellant's attorney interviewed the Examiner by telephone on January 24, 2004 in an attempt to obtain clarification of the Examiner's position. Appellant's attorney asked the Examiner specifically where, in the references, the following features of Claim 11 were disclosed: the framework that performs the steps of Claim 11, the step of determining a set of restrictions to be imposed on a customized implementation, and the concept that a wrapper instance comprises enforcement logic for enforcing those restrictions.

During the interview, the Examiner expressed his opinion that, because everything recited in Claim 11 was allegedly common knowledge, he should not have to point out with specificity exactly which reference, and exactly where in that reference, each feature of Claim 11 was allegedly disclosed. The Examiner opined that the "framework" of Claim 11 corresponds to any well-known operating system. The Examiner opined that, by virtue of instantiating a derived class that inherits from a base class and that adds methods and properties that the base class does not possess, such an operating system determines a set of restrictions to be imposed upon the base class. The Examiner opined that, by virtue of the additional methods and properties that an instance of such a derived class contains, such an instance comprises enforcement logic for enforcing the restrictions.

The merits of the Examiner's position are discussed below.

Arguments

Appellants agree that a derived class may contain methods and properties that a base class upon which the derived class is based does not contain. However, the addition of these methods and properties to the derived class does not "restrict" the base class in any way. Such methods and properties do not appear in any way to prevent the usage of functionality that was available in the base class. Actually, the addition of such methods and properties to a derived class seems to "expand" the base class, providing functionality and characteristics that the base class did not provide.

Even if the addition of such methods and properties to a derived class could be considered to somehow restrict the base class, this would not lead to the conclusion that the derived class was a wrapper class that encapsulated the base class. Inheriting from, or being derived from, a base class is not the same as encapsulating the base class. A derived class does not "encapsulate" or contain the base class from which the derived class is derived.

On page 296, Taylor discloses that legacy systems can be "absorbed" by making the legacy systems appear to be object-oriented, and that this may be accomplished by putting object-oriented wrappers around the legacy systems to allow the legacy systems to interact in an object-oriented environment. Taylor further discloses, on page 296, that wrappers can accept messages from other objects, call on conventional software to perform requested operations, and return results to the calling objects.

Even taking into account all that Taylor discloses, there is no teaching or suggestion in Taylor or any of the cited references that such legacy systems are base classes or that the wrappers are in any way classes that are derived from the legacy systems. Indeed, if the legacy systems disclosed in Taylor were object-oriented classes at all, then there would not be any need to encapsulate those legacy systems within wrappers, because instances of object-oriented classes already can interact in an object-oriented environment.

Furthermore, the wrappers discussed in Taylor do not appear in any way to "restrict" the legacy systems that those wrappers contain. Indeed, the wrappers appear to be designed to enable continued interaction with all of the features that the legacy systems originally provided. During the aforementioned interview, Applicant's attorney asked the Examiner how the wrappers could be considered to "restrict" the legacy systems, and the Examiner opined that the wrappers restricted the legacy systems because a derived class allegedly restricts a base class from which the derived class is derived. However, as is discussed above, (a) an instance of such a derived class does not "restrict" such a base class, and (b) the legacy systems contained within the wrappers are not "base classes" from which the wrappers are derived anyway.

The features recited in Claim 11 are not well known principles of object-oriented programming and design. Neither Taylor nor Pressman discloses, teaches, or suggests "determining a set of zero or more **restrictions to be imposed on said customized implementation**" or "instantiating a wrapper class to give rise to a wrapper instance, said **wrapper instance comprising enforcement logic for enforcing said restrictions**." As is discussed above, even if Taylor's legacy systems are alleged to be "customized implementations" and Taylor's "wrappers" are alleged to be "wrapper instances," the wrappers still do not comprise enforcement logic for **enforcing restrictions** on the legacy systems that they contain. The Examiner still has not demonstrated in what way Taylor's wrappers may be considered to "restrict" Taylor's legacy systems.

Based on the foregoing, it is respectfully submitted that the Examiner has failed to show that Taylor and Pressman in any way disclose or suggest at least two of the limitations of Claim 11, namely, the limitations of "determining a set of zero or more **restrictions to be imposed on said customized implementation**" and "instantiating a wrapper class to give rise to a wrapper instance, said **wrapper instance comprising enforcement logic for enforcing said restrictions**." Since the Examiner has failed to show that Taylor or Pressman, taken alone or in combination, disclose or suggest all of the limitations required by Claim 11, it is respectfully submitted that the Examiner has erred in rejecting Claim 11, and that Claim 11 is patentable over Taylor and Pressman.

In order for a *prima facie* case of obviousness to be established under 35 U.S.C. 103, the combined references must, at least when considered in combination, teach or suggest all of the limitations of the claims that are alleged to be obvious. The burden is placed upon the Examiner to make out this case by pointing out specific teachings of cited references. Such specific teachings might, if correct, constitute the evidence that is required to make out the case. However, instead of pointing out specific teachings, the Examiner has only named some references and then generally alleged that all of the limitations of Claim 11 are shown somewhere within the references, or are well known.

Appellant respectfully submits that such general allegations lack the specificity that is required to make out a *prima facie* case of obviousness. At the moment, it is not even clear which of the references the Office Action relies upon to allegedly teach or suggest particular features of Claim 11. Appellant respectfully submits that a *prima facie* case of obviousness requires more that just a bare and general assertion that everything recited in a claim is well known. If all of the features of Claim 11 are as well known as the Examiner alleges, then it should be a simple matter for the Examiner to provide specific citations to pages, paragraphs, and lines that demonstrate each feature of Claim 11 specifically. Yet, the Examiner has not done so.

Claims 12-23, 70-72, and 79-87 depend from Claim 11 and include all of the limitations of Claim 11. It is therefore respectfully submitted that Claims 12-23, 70-72, and 79-87 are patentable over Taylor and Pressman for at least the reasons set forth herein with respect to Claim 11.

Claims 34-46, 73-75, and 88-96 contain limitations similar to Claims 11-23, 70-72, and 79-87, except in the context of a framework. It is therefore respectfully submitted that Claims 11-23, 70-72, and 79-87 are patentable over Taylor and Pressman for at least the reasons set forth herein with respect to Claims 11-23, 70-72, and 79-87.

Claims 57-69, 76-78, and 97-105 contain limitations similar to Claims 11-23, 70-72, and 79-87, except in the context of a computer-readable medium. It is therefore respectfully submitted that Claims 57-69, 76-78, and 97-105 are patentable over Taylor and Pressman for at least the reasons set forth herein with respect to Claims 11-23, 70-72, and 79-87.

For at least these reasons, it is respectfully submitted that Claims 11-23, 34-46, and 57-105 are not in any way taught or suggested by Taylor or Pressman, alone or in combination, and are therefore patentable over Taylor and Pressman.

C. The Limitations of Claims 13, 14, 16, 17, 20-23, 36, 37, 39, 40, 43-46, 59, 60, 62, 63, 66-69, 81, 82, 90, 91, 99, and 100 Are Not in Any Way Taught or Suggested by Taylor, Pressman, and Shanton

Claims 13, 14, 16, 17, 20-23, 81, and 82 depend from Claim 11 and therefore include all of the distinguished limitations of Claim 11. Thus, if Claim 11 is patentable over Taylor, Pressman, and Shanton, then Claims 13, 14, 16, 17, 20-23, 81, and 82 are also patentable over Taylor, Pressman, and Shanton.

Like Taylor and Pressman, Shanton does not disclose, teach, or suggest at least the following limitations of Claim 11: "determining a set of zero or more restrictions to be imposed on said customized implementation;" and "instantiating a wrapper class to give rise to a wrapper instance, said wrapper instance comprising enforcement logic for enforcing said restrictions." Indeed, the Office Action does not even allege that Shanton discloses these features. Even assuming, arguendo, that Pressman, Taylor, and Shanton could be combined, Pressman, Taylor and Shanton still do not teach, disclose, or suggest all of the limitations of Claim 11. Thus, Claim 11 is patentable over Pressman, Taylor, and Shanton, taken individually or in combination.

Consequently, Claims 13, 14, 16, 17, 20-23, 81, and 82 are patentable over Pressman, Taylor, and Shanton, taken individually or in combination, for at least the reasons given above with reference to Claim 11.

Claims 36, 37, 39, 40, 43-46, 90, and 91 recite frameworks that comprise mechanisms for performing the methods of Claims 13, 14, 16, 17, 20-23, 81, and 82, respectively. Applicant submits that Claims 36, 37, 39, 40, 43-46, 90, and 91 are patentable over Pressman, Taylor, and Shanton for at least the reasons given above in connection with Claim 13, 14, 16, 17, 20-23, 81, and 82, respectively.

Claims 59, 60, 62, 63, 66-69, 99, and 100 recite computer-readable media that comprise instructions for causing one or more processors to perform the methods of Claims 13, 14, 16, 17, 20-23, 81, and 82, respectively. Applicant submits that Claims 59, 60, 62, 63, 66-69, 99, and 100 are patentable over Pressman, Taylor, and Shanton for at least the reasons given above in connection with Claim 13, 14, 16, 17, 20-23, 81, and 82, respectively.

IX. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted the rejections of Claims 11-23, 34-46, and 57-105 under 35 U.S.C. § 103(a) lack the requisite factual and legal bases.

Appellant therefore respectfully requests that the Honorable Board reverse the rejection of Claims 11, 12, 15, 18, 19, 34, 35, 38, 41, 42, 57, 58, 61, 64, 65, 70-80, 83-89, 92-98, and 101-105 under 35 U.S.C. § 103(a) over Taylor in view of Pressman, and the rejection of Claims 13, 14, 16, 17, 20-23, 36, 37, 39, 40, 43-46, 59, 60, 62, 63, 66-69, 81, 82, 90, 91, 99, and 100 under 35 U.S.C. § 103(a) over Taylor in view of Pressman and Shanton. This is the second time that Appellant has had to appeal the rejection of the present application to the Honorable Board; the Examiner re-opened prosecution after the last appeal. Appellant fervently hopes that this will be the last time.

Respectfully submitted,

HICKMAN PALERMO TRUONG &
BECKER LLP

Date: June 7, 2005

Christian A. Nicholes
Registration No. 50,266

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1224
Fax: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: MS Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

on June 7, 2005 by
Judy Paradoski

CLAIMS APPENDIX

11. In a system comprising an application, a framework, and an implementation class which provides an implementation for a particular service, a method performed by the framework, comprising:

receiving a request from an application for a customized implementation of a particular service;

instantiating an implementation class which provides an implementation for the particular service to give rise to an implementation instance;

determining a set of zero or more restrictions to be imposed on said customized implementation;

instantiating a wrapper class to give rise to a wrapper instance, said wrapper instance comprising enforcement logic for enforcing said restrictions;

encapsulating said implementation instance and said restrictions within said wrapper instance; and

providing said wrapper instance to the application as said customized implementation;

wherein said wrapper instance comprises one or more invocable methods, wherein said implementation instance comprises one or more invocable methods, and wherein encapsulating comprises:

mapping the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance.

12. The method of claim 11, wherein instantiating the implementation class comprises:

determining whether the implementation class is authentic; and

in response to a determination that the implementation class is authentic, instantiating the implementation class to give rise to said implementation instance.

13. The method of claim 12, wherein the implementation class has a digital signature associated therewith, and wherein determining whether the implementation class is authentic comprises:

verifying said digital signature.

14. The method of claim 12, wherein the implementation class authenticates the framework prior to giving rise to said implementation instance.

15. The method of claim 11, wherein determining the set of zero or more restrictions comprises:

accessing information specifying one or more limitations; and

processing said limitations to derive said restrictions.

16. The method of claim 15, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations.

17. The method of claim 16, wherein said default encryption limitations are derived by merging multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

18. The method of claim 11, wherein determining the set of zero or more restrictions comprises:

accessing information specifying one or more limitations;

determining permissions, if any, granted to the application; and

reconciling said limitations and said permissions to derive said restrictions.

19. The method of claim 18, wherein said limitations and said permissions are reconciled to derive restrictions which are least restrictive.

20. The method of claim 18, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations, and a set of zero or more exempt encryption limitations which apply when one or more exemption mechanisms are implemented.

21. The method of claim 20, wherein said default encryption limitations and said exempt encryption limitations are derived by merging multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

22. The method of claim 20, wherein reconciling said limitations and said permissions comprises:

determining whether the application has been granted any permissions; and

in response to a determination that the application has not been granted any permissions, deriving said restrictions from said set of default encryption limitations.

23. The method of claim 20, wherein reconciling said limitations and said permissions comprises:

determining whether the application has been granted any permissions which require implementation of a particular exemption mechanism;

in response to a determination that the application has been granted a permission which requires implementation of a particular exemption mechanism, determining whether said exempt encryption limitations allow said particular exemption mechanism to be implemented; and

in response to a determination that said exempt encryption limitations allow said particular exemption mechanism to be implemented, deriving said restrictions from said set of exempt encryption limitations.

34. In a system comprising an application and an implementation class which provides an implementation for a particular service, a framework comprising:

a mechanism for receiving a request from an application for a customized implementation of a particular service;

a mechanism for instantiating an implementation class which provides an implementation for the particular service to give rise to an implementation instance;

a mechanism for determining a set of zero or more restrictions to be imposed on said customized implementation;

a mechanism for instantiating a wrapper class to give rise to a wrapper instance, said wrapper instance comprising enforcement logic for enforcing said restrictions;

a mechanism for encapsulating said implementation instance and said restrictions within said wrapper instance; and

a mechanism for providing said wrapper instance to the application as said customized implementation;

wherein said wrapper instance comprises one or more invocable methods, wherein said implementation instance comprises one or more invocable methods, and wherein the mechanism for encapsulating comprises:

a mechanism for mapping the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance.

35. The framework of claim 34, wherein the mechanism for instantiating the implementation class comprises:

a mechanism for determining whether the implementation class is authentic; and

a mechanism for instantiating, in response to a determination that the implementation class is authentic, the implementation class to give rise to said implementation instance.

36. The framework of claim 35, wherein the implementation class has a digital signature associated therewith, and wherein the mechanism for determining whether the implementation class is authentic comprises:

a mechanism for verifying said digital signature.

37. The framework of claim 35, wherein the implementation class authenticates the framework prior to giving rise to said implementation instance.

38. The framework of claim 34, wherein the mechanism for determining the set of zero or more restrictions comprises:

a mechanism for accessing information specifying one or more limitations; and

a mechanism for processing said limitations to derive said restrictions.

39. The framework of claim 38, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations.

40. The framework of claim 39, wherein said default encryption limitations are derived by merging multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

41. The framework of claim 34, wherein the mechanism for determining the set of zero or more restrictions comprises:

a mechanism for accessing information specifying one or more limitations;

a mechanism for determining permissions, if any, granted to the application; and

a mechanism for reconciling said limitations and said permissions to derive said restrictions.

42. The framework of claim 41, wherein said limitations and said permissions are reconciled to derive restrictions which are least restrictive.

43. The framework of claim 41, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations, and a set of zero or more exempt encryption limitations which apply when one or more exemption mechanisms are implemented.

44. The framework of claim 43, wherein said default encryption limitations and said exempt encryption limitations are derived by merging multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

45. The framework of claim 43, wherein the mechanism for reconciling said limitations and said permissions comprises:

a mechanism for determining whether the application has been granted any permissions; and

a mechanism for deriving, in response to a determination that the application has not been granted any permissions, said restrictions from said set of default encryption limitations.

46. The framework of claim 43, wherein the mechanism for reconciling said limitations and said permissions comprises:

a mechanism for determining whether the application has been granted any permissions which require implementation of a particular exemption mechanism;

a mechanism for determining, in response to a determination that the application has been granted a permission which requires implementation of a particular exemption mechanism, whether said exempt encryption limitations allow said particular exemption mechanism to be implemented; and

a mechanism for deriving, in response to a determination that said exempt encryption limitations allow said particular exemption mechanism to be implemented, said restrictions from said set of exempt encryption limitations.

57. In a system comprising an application and an implementation class which provides an implementation for a particular service, a computer readable medium having stored thereon instructions which, when executed by one or more processors, cause the one or more processors to implement a framework which dynamically constructs a customized implementation, said computer readable medium comprising:

instructions for causing one or more processors to receive a request from an application for a customized implementation of a particular service;

instructions for causing one or more processors to instantiate an implementation class which provides an implementation for the particular service to give rise to an implementation instance;

instructions for causing one or more processors to determine a set of zero or more restrictions to be imposed on said customized implementation;

instructions for causing one or more processors to instantiate a wrapper class to give rise to a wrapper instance, said wrapper instance comprising enforcement logic for enforcing said restrictions;

instructions for causing one or more processors to encapsulate said implementation instance and said restrictions within said wrapper instance; and

instructions for causing one or more processors to provide said wrapper instance to the application as said customized implementation;

wherein said wrapper instance comprises one or more invocable methods, wherein said implementation instance comprises one or more invocable methods, and wherein the instructions for causing one or more processors to encapsulate comprises:

instructions for causing one or more processors to map the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance.

58. The computer readable medium of claim 57, wherein the instructions for causing one or more processors to instantiate the implementation class comprises:

instructions for causing one or more processors to determine whether the implementation class is authentic; and

instructions for causing one or more processors to instantiate, in response to a determination that the implementation class is authentic, the implementation class to give rise to said implementation instance.

59. The computer readable medium of claim 58, wherein the implementation class has a digital signature associated therewith, and wherein the instructions for causing one or more processors to determine whether the implementation class is authentic comprises:

instructions for causing one or more processors to verify said digital signature.

60. The computer readable medium of claim 58, wherein the implementation class authenticates the framework prior to giving rise to said implementation instance.

61. The computer readable medium of claim 57, wherein the instructions for causing one or more processors to determine the set of zero or more restrictions comprises:

instructions for causing one or more processors to access information specifying one or more limitations; and

instructions for causing one or more processors to process said limitations to derive said restrictions.

62. The computer readable medium of claim 61, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations.

63. The computer readable medium of claim 62, wherein said default encryption limitations are derived by merging multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

64. The computer readable medium of claim 57, wherein the instructions for causing one or more processors to determine the set of zero or more restrictions comprises:

instructions for causing one or more processors to access information specifying one or more limitations;

instructions for causing one or more processors to determine permissions, if any, granted to the application; and

instructions for causing one or more processors to reconcile said limitations and said permissions to derive said restrictions.

65. The computer readable medium of claim 64, wherein said limitations and said permissions are reconciled to derive restrictions which are least restrictive.

66. The computer readable medium of claim 64, wherein the particular service is an encryption/decryption service, and wherein said information comprises a set of one or more default encryption limitations, and a set of zero or more exempt encryption limitations which apply when one or more exemption mechanisms are implemented.

67. The computer readable medium of claim 66, wherein said default encryption limitations and said exempt encryption limitations are derived by merging

multiple jurisdiction policies and extracting therefrom the most restrictive encryption limitations.

68. The computer readable medium of claim 66, wherein the instructions for causing one or more processors to reconcile said limitations and said permissions comprises:

instructions for causing one or more processors to determine whether the application has been granted any permissions; and

instructions for causing one or more processors to derive, in response to a determination that the application has not been granted any permissions, said restrictions from said set of default encryption limitations.

69. The computer readable medium of claim 66, wherein the instructions for causing one or more processors to reconcile said limitations and said permissions comprises:

instructions for causing one or more processors to determine whether the application has been granted any permissions which require implementation of a particular exemption mechanism;

instructions for causing one or more processors to determine, in response to a determination that the application has been granted a permission which requires implementation of a particular exemption mechanism, whether said exempt encryption limitations allow said particular exemption mechanism to be implemented; and instructions for causing one or more processors to derive, in response to a determination that said exempt encryption limitations allow said particular exemption mechanism to be implemented, said restrictions from said set of exempt encryption limitations.

70. The method of claim 11, wherein determining said set of zero or more restrictions includes determining a set of zero or more restrictions that are specific to said application.

71. The method of claim 70, wherein determining said set of zero or more restrictions that are specific to said application includes determining a set of zero or more restrictions that are customized for said application.

72. The method of claim 11, wherein said set is a first set, and wherein said customized implementation is a first customized implementation, and further comprising:

receiving a request from a second application for a second customized implementation of said particular service, wherein said second customized implementation differs from said first customized implementation;

instantiating said implementation class which provides said implementation for said particular service to give rise to a second implementation instance;

determining a second set of zero or more restrictions to be imposed on said second customized implementation, wherein said second set differs from said first set;

instantiating said wrapper class to give rise to a second wrapper instance, said second wrapper instance comprising enforcement logic for enforcing said second set of zero or more restrictions;

encapsulating said second implementation instance and said second set of zero or more restrictions within said second wrapper instance; and

providing said second wrapper instance to said second application as said second customized implementation.

73. The framework of claim 34, wherein said mechanism for determining said set of zero or more restrictions includes a mechanism for determining a set of zero or more restrictions that are specific to said application.

74. The framework of claim 73, wherein said mechanism for determining said set of zero or more restrictions that are specific to said application includes a mechanism for determining a set of zero or more restrictions that are customized for said application.

75. The framework of claim 34, wherein said set is a first set, and wherein said customized implementation is a first customized implementation, and further comprising:

a mechanism for receiving a request from a second application for a second customized implementation of said particular service, wherein said second customized implementation differs from said first customized implementation;

a mechanism for instantiating said implementation class which provides said implementation for said particular service to give rise to a second implementation instance;

a mechanism for determining a second set of zero or more restrictions to be imposed on said second customized implementation, wherein said second set differs from said first set;

a mechanism for instantiating said wrapper class to give rise to a second wrapper instance, said second wrapper instance comprising enforcement logic for enforcing said second set of zero or more restrictions;

a mechanism for encapsulating said second implementation instance and said second set of zero or more restrictions within said second wrapper instance; and

a mechanism for providing said second wrapper instance to said second application as said second customized implementation.

76. The computer readable medium of claim 57, wherein said instructions for determining said set of zero or more restrictions include instructions for determining a set of zero or more restrictions that are specific to said application.

77. The computer readable medium of claim 76, wherein said instructions for determining said set of zero or more restrictions that are specific to said application include instructions for determining a set of zero or more restrictions that are customized for said application.

78. The computer readable medium of claim 57, wherein said set is a first set, and wherein said customized implementation is a first customized implementation, and further comprising:

instructions for receiving a request from a second application for a second customized implementation of said particular service, wherein said second customized implementation differs from said first customized implementation;

instructions for instantiating said implementation class which provides said implementation for said particular service to give rise to a second implementation instance;

instructions for determining a second set of zero or more restrictions to be imposed on said second customized implementation, wherein said second set differs from said first set;

instructions for instantiating said wrapper class to give rise to a second wrapper instance, said second wrapper instance comprising enforcement logic for enforcing said second set of zero or more restrictions;

instructions for encapsulating said second implementation instance and said second set of zero or more restrictions within said second wrapper instance; and

instructions for providing said second wrapper instance to said second application as said second customized implementation.

79. The method of claim 11, wherein said wrapper instance is invocable by the application without further interaction with the framework.

80. The method of claim 11, wherein the implementation class provides an unrestricted implementation for the particular service.

81. The method of claim 80, wherein the particular service is an encryption/decryption service, and wherein the unrestricted implementation provided by the implementation class is capable of using unlimited encryption/decryption parameters.

82. The method of claim 81, wherein the unrestricted implementation provided by the implementation class is capable of using encryption/decryption keys of any size.

83. The method of claim 11, wherein said enforcement logic enforces said restrictions on said implementation instance.

84. The method of claim 83, wherein said enforcement logic enforces said restrictions on said implementation instance by:

receiving a set of desired parameters from the application;

determining whether the desired parameters exceed said restrictions; and

in response to a determination that the desired parameters exceed said restrictions, preventing said implementation instance from operating.

85. The method of claim 84, wherein said enforcement logic is invoked upon initialization of said wrapper instance.

86. The method of claim 11, wherein the system further comprises an exemption mechanism class which provides an implementation for a particular exemption mechanism, and wherein said method further comprises:

instantiating the exemption mechanism class to give rise to an exemption mechanism instance; and

encapsulating said exemption mechanism instance within said wrapper instance.

87. The method of claim 86, wherein said enforcement logic is invoked upon initialization of said wrapper instance, and when invoked, said enforcement logic:

determines whether said exemption mechanism instance has been invoked; and

in response to a determination that said exemption mechanism instance has not been invoked, preventing said implementation instance from operating.

88. The framework of claim 34, wherein said wrapper instance is invocable by the application without further interaction with the framework.

89. The framework of claim 34, wherein the implementation class provides an unrestricted implementation for the particular service.

90. The framework of claim 89, wherein the particular service is an encryption/decryption service, and wherein the unrestricted implementation provided by the implementation class is capable of using unlimited encryption/decryption parameters.

91. The framework of claim 90, wherein the unrestricted implementation provided by the implementation class is capable of using encryption/decryption keys of any size.

92. The framework of claim 34, wherein said enforcement logic enforces said restrictions on said implementation instance.

93. The framework of claim 92, wherein said enforcement logic enforces said restrictions on said implementation instance by:

receiving a set of desired parameters from the application;

determining whether the desired parameters exceed said restrictions; and

in response to a determination that the desired parameters exceed said restrictions, preventing said implementation instance from operating.

94. The framework of claim 93, wherein said enforcement logic is invoked upon initialization of said wrapper instance.

95. The framework of claim 34, wherein the system further comprises an exemption mechanism class which provides an implementation for a particular exemption mechanism, and wherein said framework further comprises:

a mechanism for instantiating the exemption mechanism class to give rise to an exemption mechanism instance; and a mechanism for encapsulating said exemption mechanism instance within said wrapper instance.

96. The framework of claim 95, wherein said enforcement logic is invoked upon initialization of said wrapper instance, and when invoked, said enforcement logic:

determines whether said exemption mechanism instance has been invoked; and

in response to a determination that said exemption mechanism instance has not been invoked, preventing said implementation instance from operating.

97. The computer readable medium of claim 57, wherein said wrapper instance is invocable by the application without further interaction with the framework.

98. The computer readable medium of claim 57, wherein the implementation class provides an unrestricted implementation for the particular service.

99. The computer readable medium of claim 98, wherein the particular service is an encryption/decryption service, and wherein the unrestricted implementation provided by the implementation class is capable of using unlimited encryption/decryption parameters.

100. The computer readable medium of claim 99, wherein the unrestricted implementation provided by the implementation class is capable of using encryption/decryption keys of any size.

101. The computer readable medium of claim 57, wherein said enforcement logic enforces said restrictions on said implementation instance.

102. The computer readable medium of claim 101, wherein said enforcement logic enforces said restrictions on said implementation instance by:

receiving a set of desired parameters from the application;

determining whether the desired parameters exceed said restrictions; and

in response to a determination that the desired parameters exceed said restrictions, preventing said implementation instance from operating.

103. The computer readable medium of claim 102, wherein said enforcement logic is invoked upon initialization of said wrapper instance.

104. The computer readable medium of claim 57, wherein the system further comprises an exemption mechanism class which provides an implementation for a particular exemption mechanism, and wherein said computer readable medium further comprises:

instructions for causing one or more processors to instantiate the exemption mechanism class to give rise to an exemption mechanism instance; and

instructions for causing one or more processors to encapsulate said exemption mechanism instance within said wrapper instance.

105. The computer readable medium of claim 104, wherein said enforcement logic is invoked upon initialization of said wrapper instance, and when invoked, said enforcement logic:

determines whether said exemption mechanism instance has been invoked; and

in response to a determination that said exemption mechanism instance has not been invoked, preventing said implementation instance from operating.